



dCache Configuration and administration

Zhenping (Jane) Liu

ATLAS Computing Facility, Physics Department
Brookhaven National Lab

09/12 – 09/13, 2005 USATLAS Tier-1 & Tier-2 dCache Workshop



Agenda

- PNFS filesystem setup
- Pool manager configuration
- Monitoring
- Miscellaneous stuff
- Useful document links



Pnfs Filesystem Setup





PNFS

- Data files do not actually reside in the /pnfs namespace. Errors occur on attempts to read or write the content of the files, or to manipulate the content.
 - You have to use client tools like dccp, globus-url-copy, srmcp to read/write data into dCache. Unix command “cp” can NOT be used to read/write real data.
- Virtually any non-I/O UNIX command can be used in the /pnfs namespace.
- Commands that can NOT be used in PNFS space:
 - cp, mv, cat, more, less, grep, head, tail, wc, od, file.
- Commands that can be used in PNFS space:
 - ls, pwd, find, rm and rmdir, cd, mkdir, ln (hard links only), chown, chmod.
- **Do NOT use "mv" to move files and directories in PNFS area.**
- Please refer to http://www.atlasgrid.bnl.gov/dcache/manuals/pnfs_unixcommands.html



PNFS

- The primary configuration file for pnfs is `/usr/etc/pnfsSetup`.
- The log files for the three pnfs server daemons are `/var/log/pmountd.log`, `/var/log/dbserver.log`, and `/var/log/pnfsd.log`.



PNFS

- PNFS database files:
/opt/pnfsdb/pnfs/databases/.
 - The pnfs system finds them via the information in the directory /opt/pnfsdb/pnfs/info/.
- Each database file is handled by one **dbserver** daemon and each access will lock the database file.
- Each database file/server is the container for one directory sub-tree.



The Databases of pnfs

- pnfs stores all the information in gdbm database files.
 - Since each operation will lock the database file used globally and since GNU dbm cannot handle database files larger than 2GB, it is advisable to “split” them suitably to future usage.
 - Each database stores the information of a sub-tree of the pnfs filesystem namespace. Which database is responsible for a directory and subsequent subdirectories is determined at creation time of the directory.
- Each database is handled by a separate server process. The maximum number of servers is set by the variable `shmservers` in file `/usr/etc/pnfsSetup`.



PNFS

■ pnfs-IDs

- Each file in pnfs has a unique 12 byte long pnfs ID.
 - Comparable to the inode number in other filesystems.
- The pnfs ID used for a file will never be reused, even if the file is deleted.
- dCache uses the pnfs ID for all internal references to a file
 - `cat /pnfs/usatlas.bnl.gov/data/zhliu/".(id)(test.dat)"`
00010000000000000002320B8
 - On a pool which has the data copy for
`/pnfs/usatlas.bnl.gov/data/zhliu/test.dat`, the copy is named
00010000000000000002320B8



PNFS --- Directory Tags

- In the pnfs filesystem, each directory has a number of tags.

The existing tags may be listed with

```
[user] $ cat '.(tags)()'
```

```
.(tag)(OSMTemplate)
```

```
.(tag)(sGroup)
```

and the content of a tag can be read with

```
[user] $ cat '.(tag)(OSMTemplate)'
```

```
StoreName myStore
```

```
[user] $ grep "" $(cat ".(tags)()")
```

```
.(tag)(OSMTemplate):StoreName myStore
```

```
.(tag)(sGroup):STRING
```



PNFS --- Directory Tags

- OSMTemplate specifies the name of the store used by dCache to construct the *storage class* .
- sGroupThe storage group is also used to construct the *storage Class*
- Tags are *inherited* from the parent directory by a newly created directory.



Global Configuration with Wormholes

- Global Configuration with [Wormholes](#)
- A way to distribute configuration information to all directories in the pnfs filesystem. It can be accessed in a subdirectory `.(config)()` of any pnfs-directory. It behaves similar to a hardlink.
- In the default configuration, this link points to `/pnfs/fs/admin/etc/config/`. In it are three files:
 - `.(config)()/serverId` contains the domain name of the site,
 - `.(config)()/serverName` the fully qualified name of the pnfs server, and
 - `.(config)()/serverRoot` should contain “0000000000000000000000001080 .”.
- The dCache specific configuration can be found in `.(config)()/dCache/dcache.conf`. This file contains dCap doors which may be used by dCap clients when not using URLs. The **dccp** program will choose randomly between the doors listed here. (Useful when you need add dcap doors)
- *** Note: please use “echo” to modify/create files under `/pnfs/fs/admin/...` Otherwise, you may get zero size file. Anyway, every time after you change files under `/pnfs/fs/admin/...`, do check the size.



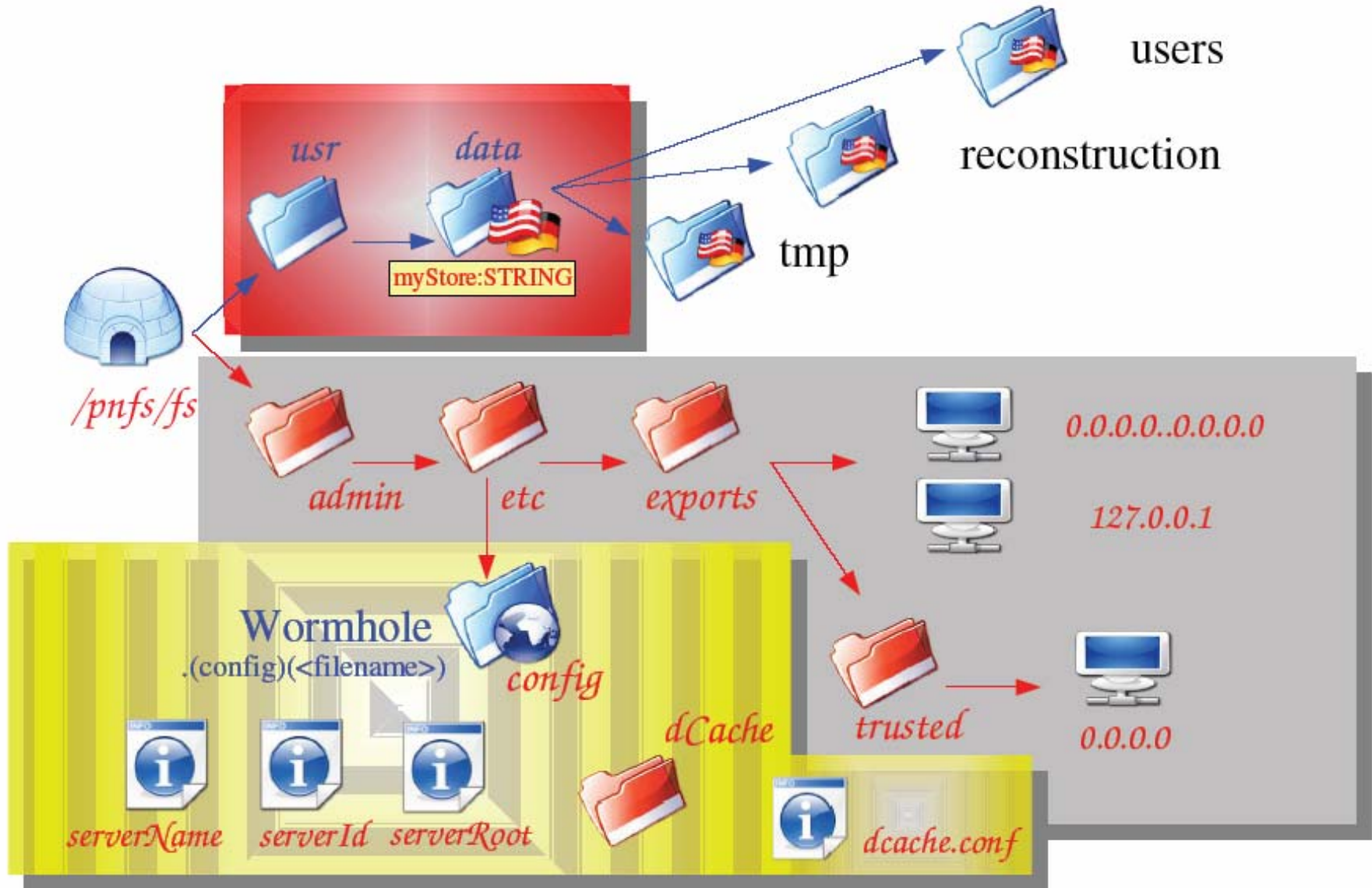
Deleted Files in pnfs

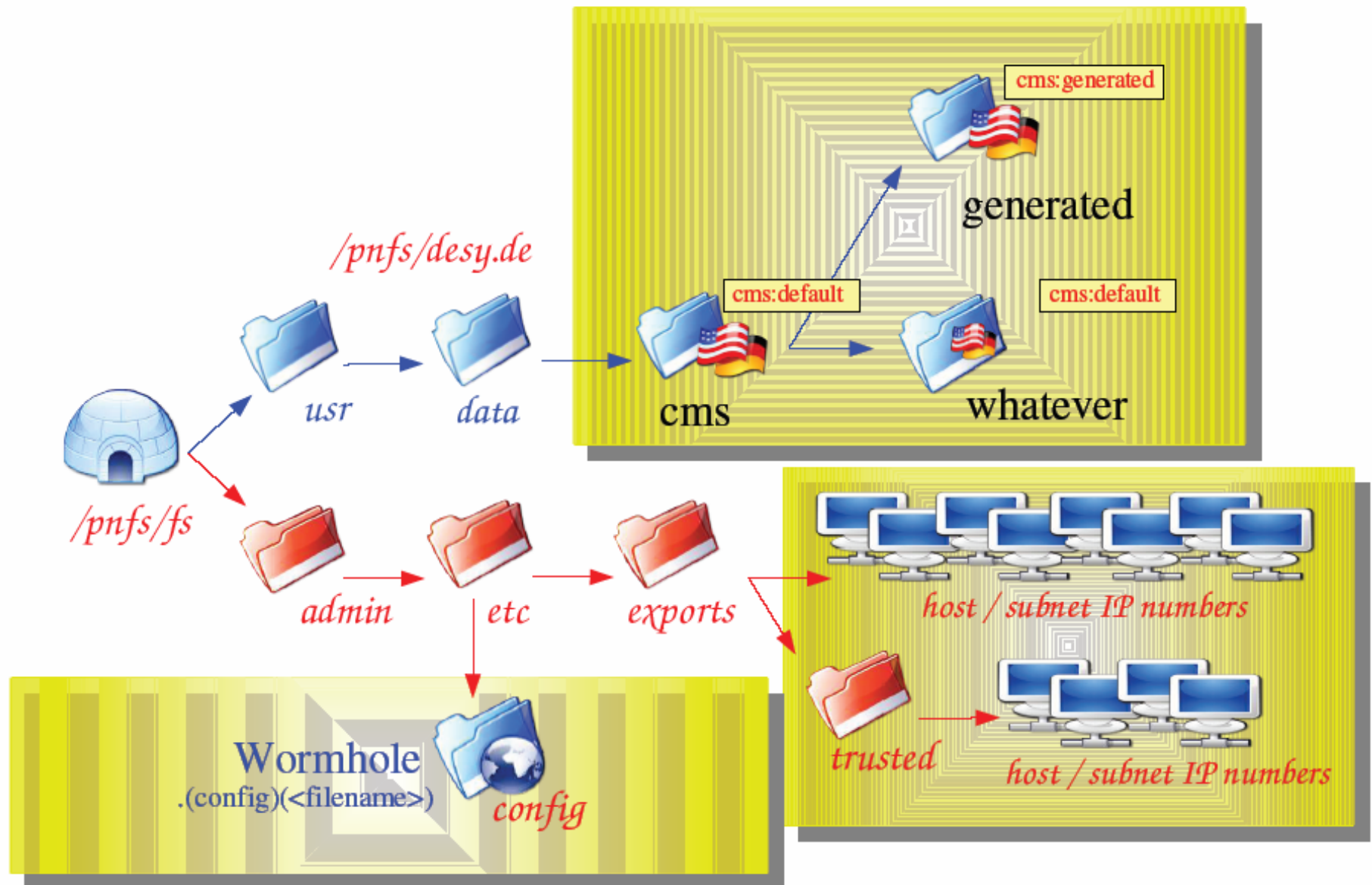
- When a file in the pnfs filesystem is deleted the server stores information about it in the subdirectories of `/opt/pnfsdb/pnfs/trash/`.
- For dCache, the cleaner cell in the pnfsDomain is responsible for deleting the actual files from the pools asynchronously. It uses the files in the directory `/opt/pnfsdb/pnfs/trash/2/`.



Access Control

- `/pnfs/fs/admin/etc/exports/<hostIP>` and `/pnfs/fs/admin/etc/exports/<netMask>..<netPart>` are used to control the host-based access to the pnfs filesystem via mount points.
 - In the initial configuration there is one file `/pnfs/fs/admin/etc/exports/0.0.0.0..0.0.0.0` containing
`/pnfs /0/root/fs/usr/ 30 nooptions`
thereby allowing all hosts to mount the part of the pnfs filesystem containing the user data.
 - There also is a file `/pnfs/fs/admin/etc/exports/127.0.0.1` containing
`/fs /0/root/fs 0 nooptions`
`/admin /0/root/fs/admin 0 nooptions`
The first line is the mountpoint used by the admin node.







pool manager

- Deciding how to handle every request.
 - Which pool to use
- Two important sub-modules: the pool selection unit (PSU) and the cost manager (CM).
 - The PSU is responsible for finding the pools allowed to use for a specific transfer-request. From those the CM selects the optimal one.



The PoolManager

Pool Selection Unit

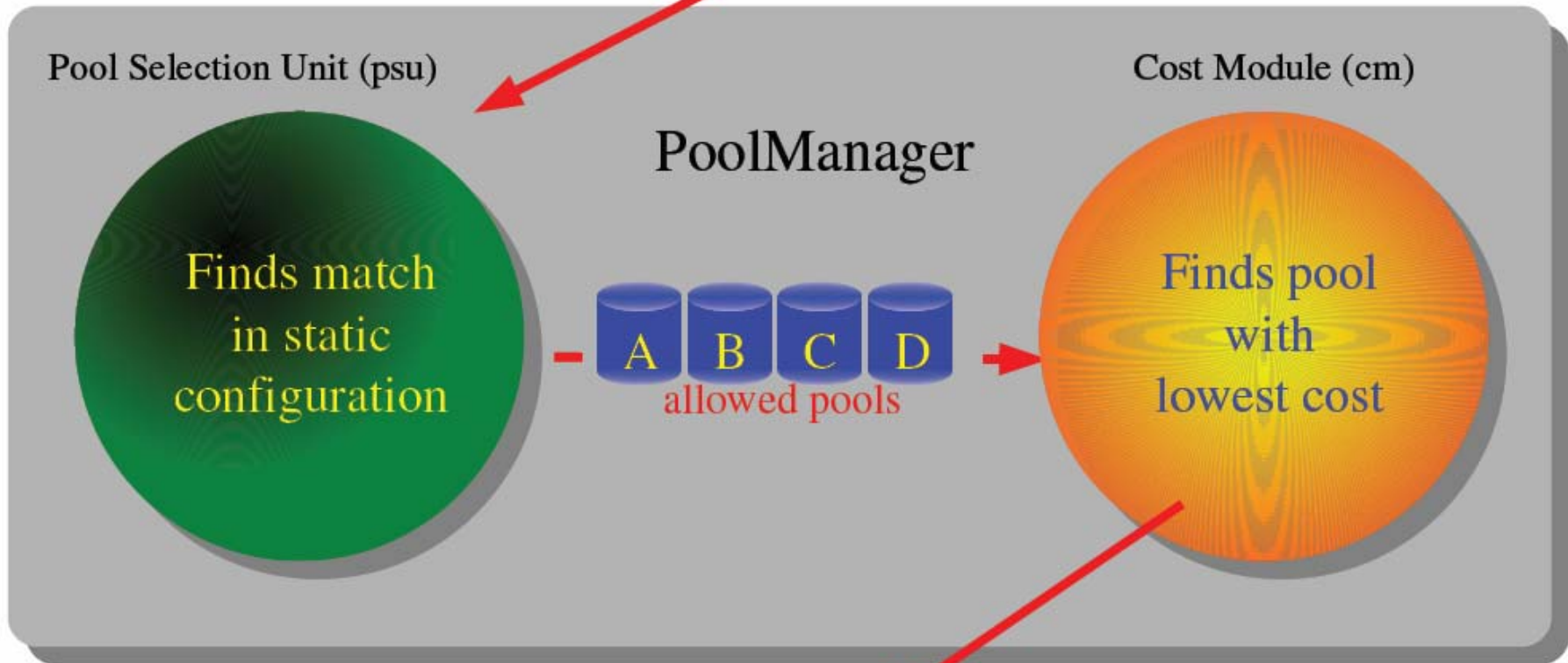
Cost Module





Incoming Request

farmnode-111.desy.de IP Storage Class cms:generated I/O Direction GET



Selected Pool







IP
farmnode-111.desy.de








Storage Class
cms:generated

I/O Direction
GET

Incoming Request

!!! tries to find match in attraction table !!!

Netmask	Group of Storage Classes	I/O	Pref	LINK	Group Of Pools
*.desy.de	 cms:generated cms:raw	GET	10	→	A B C D
*.desy.de	 cms:mc2004 cms:mc2005	GET	10	→	E F
*.desy.de	 *	GET	5	→	A G H
*	 cms:generated cms:raw	GET	10	→	M N K
*.desy.de	 cms:generated cms:users	GET	0	→	W X

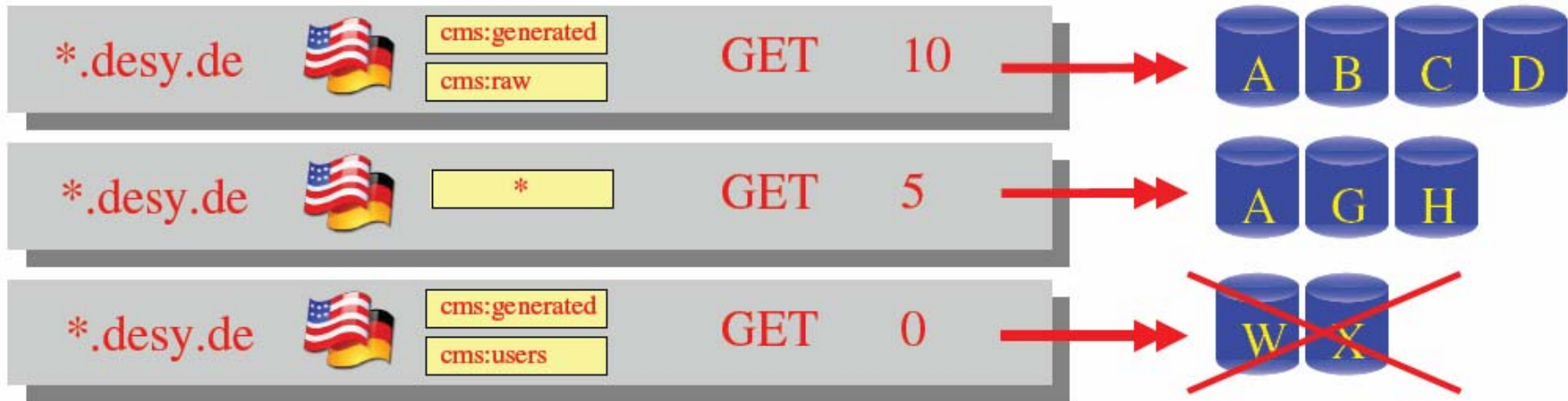




 IP
farmnode-111.desy.de
  Storage Class
cms:generated
  I/O Direction
GET

Incoming Request

!!! Result sorted by preference !!!



Selected Pool Candidates :





```
psu create pool A
psu create pool B
psu create pool C
psu create pool D
```

```
psu create pgroup ABCD-pools
psu addto pgroup ABCD-pools A
psu addto pgroup ABCD-pools B
psu addto pgroup ABCD-pools C
psu addto pgroup ABCD-pools D
```

*.desy.de



cms:generated

cms:raw

```
psu create unit -store cms:generated@osm
psu create unit -store cms:raw@osm
psu create ugroup gen_and_raw
psu addto ugroup \
    gen_and_raw cms:generated@osm
psu addto ugroup \
    gen_and_raw cms:raw@osm
psu create unit -net 255.255.0.0/131.169.0.0
psu create ugroup desy-net
psu addto ugroup \
    desy-net 255.255.0.0/131.169.0.0
```

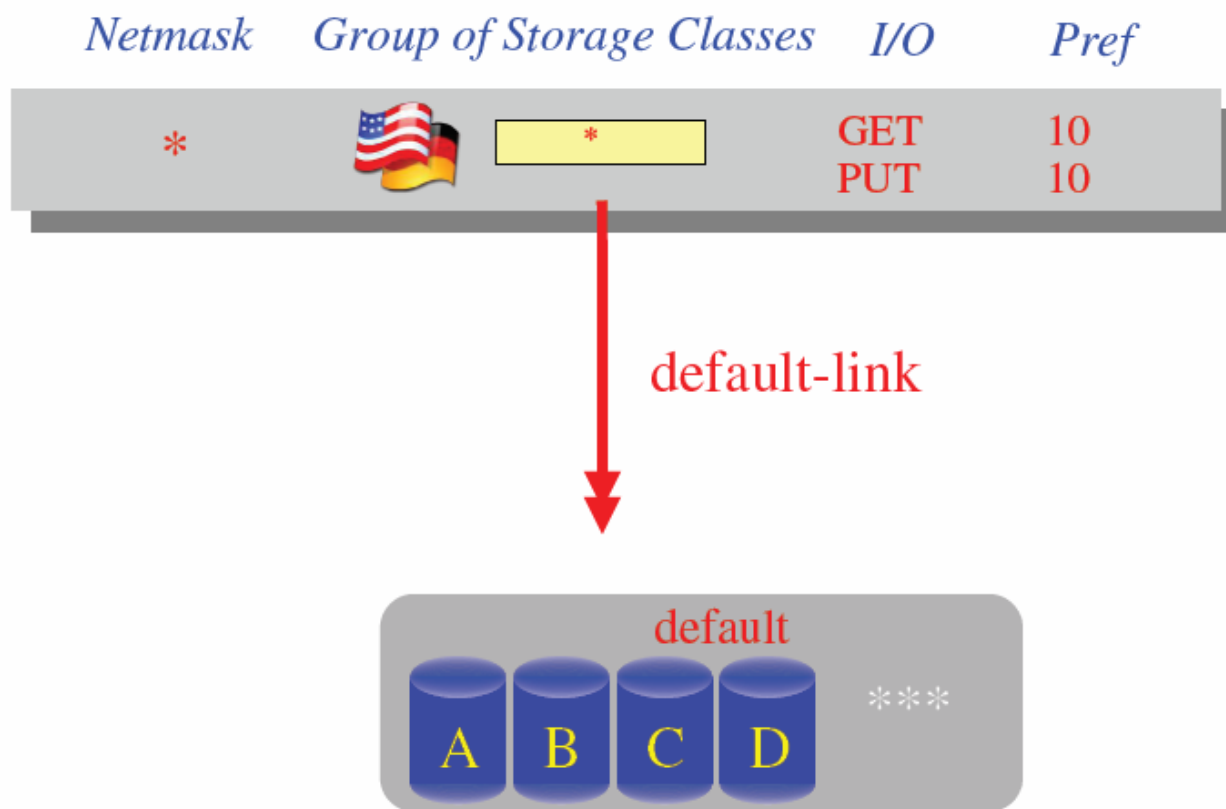
```
psu create link raw-gen-from-desy gen_and_raw desy-net
psu add link raw-gen-from-desy ABCD-pools
psu set link raw-gen-from-desy -readpref=10 -writepref=20 -cachepref=0
```





pool manager

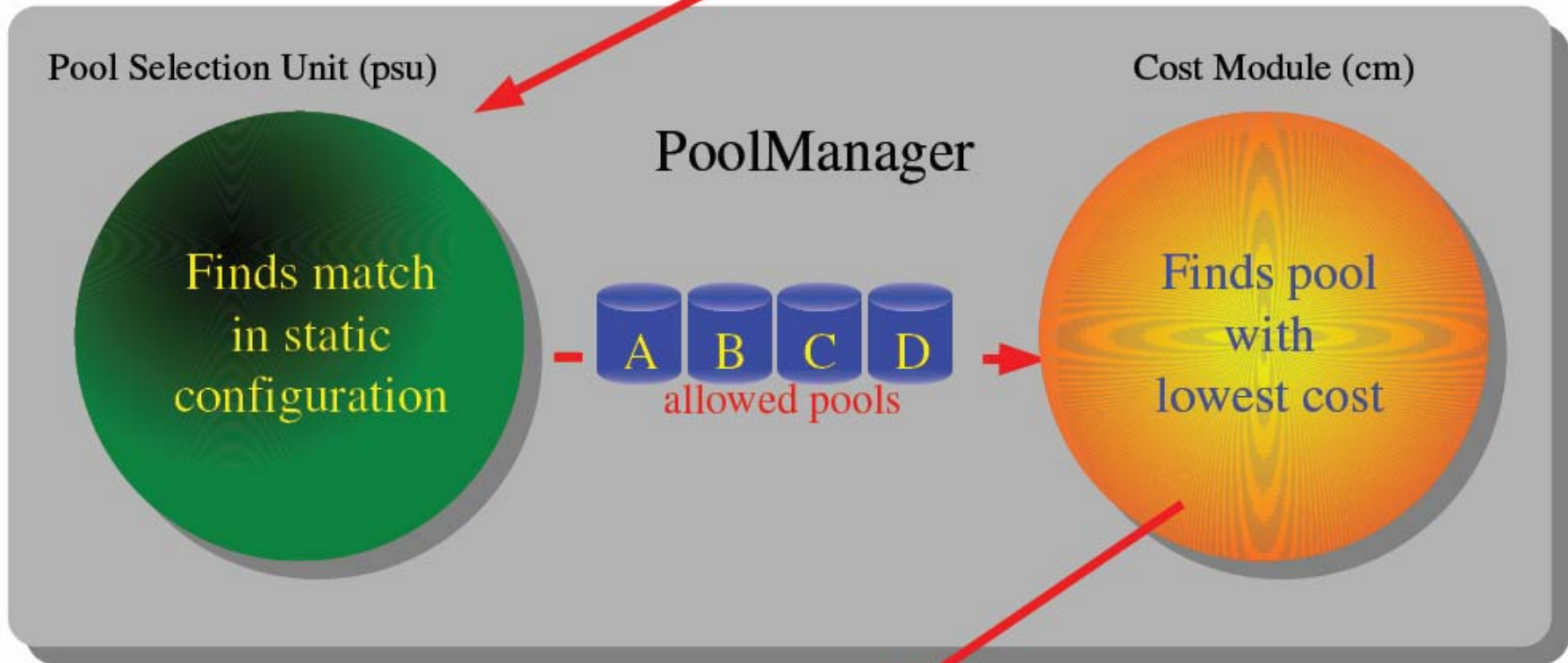
- Highly configurable.
- Dividing pools according to Storage group, network subnet mask, I/O direction.
 - Storage group: different VO, or different type of data (raw or generated)
 - network subnet mask: different subnets
 - I/O direction: Write pools or read pools
 - Write pool: e.g.,
 - -readpref=10 -writepref=0 -cachepref=10
 - Read pool: e.g.
 - -readpref=0 -writepref=10 -cachepref=0





Incoming Request

farmnode-111.desy.de IP Storage Class cms:generated I/O Direction GET



Selected Pool





$$\text{Mover Cost (CC)} = \frac{\sum_{\text{Mover Queues}} \frac{\text{active} + \text{waiting}}{\text{maximum allowed movers}}}{\text{Number of Mover Queues}}$$

Mover Queues

*from tape to disk***rh set max active <num>***from disk to tape***st set max active <num>***pool to pool (server)***p2p set max active <num>***pool to pool (client)***pp set max active <num>***client to/from pool***mover -queue=<queueName> set max active <num>**

$$\text{Space Cost (SC)} = 1 + \frac{1}{\text{LRU file}} * 3600 * 24 * 7 * \text{<breakEven>}$$

Break Even : **set breakeven <breakEven>**



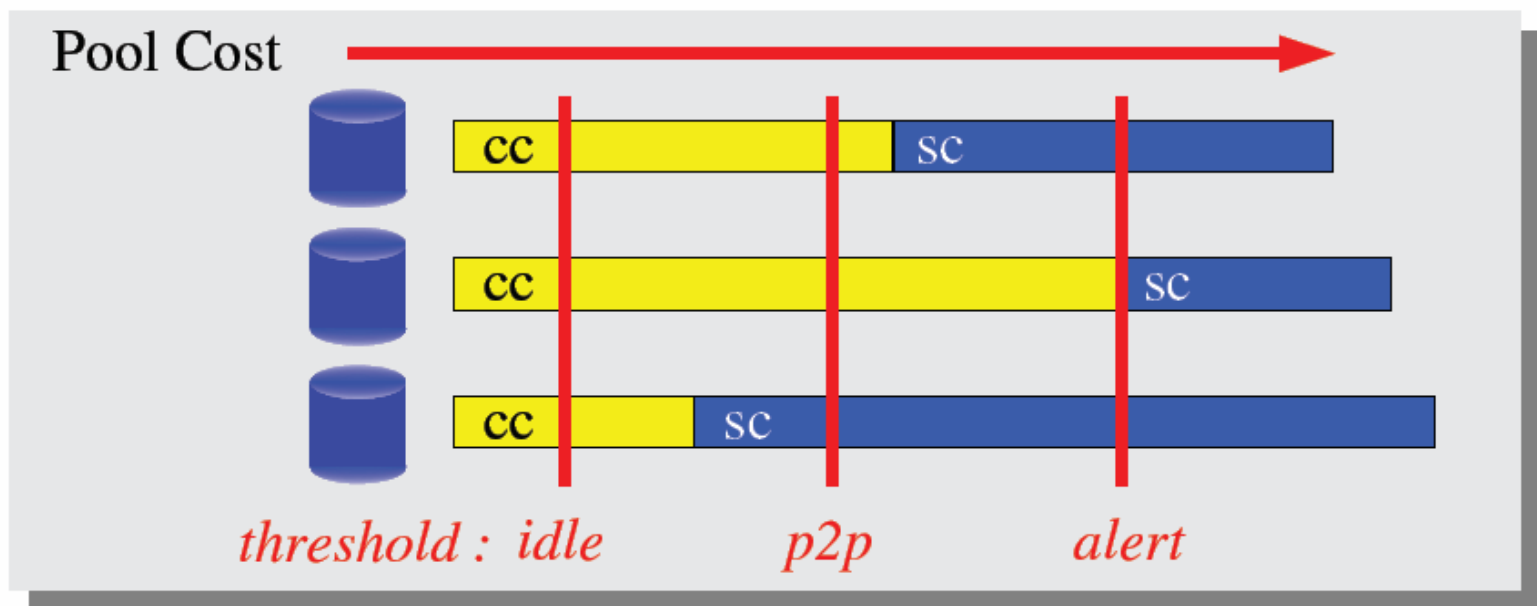
Total Cost per pool (tc) :

$tc = \text{cpu-factor} * cc + \text{space-factor} * sc$
for **write/p2p-server/restore**

$tc = \text{cpu-factor} * cc$
for **read/p2p-client**

setting factors : **set pool decision -spacefactor=<factor> -cpufactor=<factor>**





setting idle threshold : **set costcuts -idle=<idle-threshold>**
setting p2p threshold : **set costcuts -p2p=<p2p-threshold>**
setting alert threshold : **set costcuts -alert=<alert-threshold>**

idle : below : don't do load balancing on multiple file copies
p2p : above : start p2p transfers on high load of best pool
alert : above : stop p2p transfers on high load, do restore instead





pool manager

- Config/PoolManager.conf
- Two ways:
 - When system is running, modify PoolManager.conf and “reload –yes” in admin shell
 - Recommend this. Neat.
 - Run cell commands and save (all from admin shell)
 - The order of pool names is in a mess.
- Refer to dCache book
 - Chapter 5. Configuring the Pool Manager at http://www.dcache.org/manuals/Book/cf_pn.shtml



The Web Interface for Monitoring

- Link: <http://<adminNode>:2288/>
- Cell Services:
 - status of some important cells
- Pool Usage
 - the current space usage on each pool
- Pool Request Queues
 - The number current requests handled by each pool.
- Actions Log
 - keeps track of all the transfers performed by the pools up to now.
- Pools
 - the current configuration of the pool selection unit in the pool manager



The Web Interface for Monitoring

- Transfer knowledge, Active transfers, Active transfer details
 - Status of current transfers
- Restore queue, Lazy restore queue
 - Status of P2p or staging requests
- *** Note: the default setup doesn't export all important cells and menu items as above,
 - Please refer to “The Web Interface for Monitoring dCache ” at http://www.atlasgrid.bnl.gov/dcacheladmin/installation_v1.6.5.3_gdbm.htm#admin_monitor for how to do it.



Admin interface

- **Powerful administration interface**
- **Access interactively**
 - `ssh -c blowfish -p 22223 -l admin`
<adminNode>
- **Access non-interactively by scripts**
 - Using public/private key.
 - Very useful for maintenance job.



Billing

- The raw information about all dCache activities can be found in `billing/<YYYY>/<MM>/billing-<YYYY.MM.DD>`.

```
05.31 22:35:16 [pool:<pool-name>:transfer]  
[0001000000000000000000001320,24675] myStore:STRING@osm  
24675 474 true {GFtp-1.0 <client-host-fqn> 37592} {0:""}
```

The first barcket contains the pool name, the second the [*PNFS Id*](#) and the size of the file which is transferred. Then follows the [*storage class*](#), the actual amount of bytes transferred, and the number of milliseconds the transfer took. The next entry is true if the transfer was a wrote data to the pool. The first braces contain the protocol, client FQN, and the client host data transfer listen port. The final bracket contains the return status and a possible error message.



Important cell commands

- Common

- info - Print info about the cell.

- help –print available commands about the cell



Important cell commands (cont.)

■ PnfsManager

- `pnfsidof` - Print the PNFS id of a file given by its global path.
- `pathfinder` - Print the global or local path of a file from its PNFS id.
- `storageinfoof` - Print the storage info of a file.
- `cacheinfoof` - Print the cache info of a file.



Important cell commands (cont.)

■ Pool

- rep ls - List the files currently in the repository of the pool.
- st set max active - Set the maximum number of active store transfers.
- rh set max active - Set the maximum number of active restore transfers.
- mover set max active - Set the maximum number of active client transfers.
- p2p set max active - Set the maximum number of active pool-to-pool server transfers.
- pp set max active - Set the value used for scaling the performance cost of pool-to-pool client transfers analogous to the other set max active-commands.
- set gap - Set the gap parameter - the size of free space below which it will be assumed that the pool is full within the cost calculations.
- set breakeven - Set the breakeven parameter - used within the cost calculations.
- mover ls - List the active and waiting client transfer requests.



Important cell commands (cont.)

■ PoolManager

- rc ls - List the requests currently handled by the Pool Manager
- cm ls - List information about the pools in the cost module cache.
- set pool decision - Set the factors for the calculation of the total costs of the pools.



Log files

- Very useful for debugging problem.
 - When it's not enough, increase debug level
 - E.g., in /opt/etcd/etcd/config/gridftpdoor.batch, if “set **printout 2**”, you don't have info about grid DN of a user who is requesting files; However, “set **printout 3**”, you can have the info.



Stop a pool gracefully

- To take a pool out of service
 - issue commands first
 - pool disable
 - mover set max active 0
 - rh set max active 0
 - st set max active 0
 - p2p set max active 0
 - Then wait until all active jobs in queues are zero and then shut down the pool
 - Note: sometime some jobs just don't finish due to hang-up. Need double check and just kill it.



Stop a door gracefully

- To take a door out of service

- **drain** the **door** first

- set max logins 0

- Wait until active login is zero and then shut down

- Note: sometime some login just can't be closed due to issues. Need double check and just stop the door if that's the case.



Start/Stop dCache system

- Start: PNFS, core, pools/doors
- Stop: pools/doors, core, PNFS



PNFS mount

- Suggest to use automount

- Using mount way in dCache installation script, if some user is accessing PNFS, stopping PNFS will cause problem (sometimes one has to reboot server)
- Automounting is the process where mounting and unmounting of certain filesystems is done automatically by a daemon. If the filesystem is unmounted, and a user attempts to access it, it will be automatically (re)mounted. This is especially useful in large networked environments



Other stuff

- PNFS database backup
- Set up log rotation
- Add automatic dCache startup
- Add automatic Postgres startup
- Vacuum (SRM) Postgres DB periodically
- Add Crontab job to generate kpwd from grid-mapfile
- How to deal with client getting "Too many open files" problem
 - Set these two parameters large
 - ulimit -n 65536 (The maximum number of open file descriptors.)
 - echo "8192" >/proc/sys/fs/file-max
 - Set them as default on door nodes
- Make sure PNFS_LOG directory is existed



Useful Document links from Developers

- **dCache project web site**
 - <http://www.dcache.org/>
- **dCache Book (*****)**
 - <http://www.dcache.org/manuals/Book/>
- **dCache Guide for LCG Site Administrators**
 - <http://wiki.gsi.de/pub/Gridadmin/ToDo2005Apr21080327/dCache4SiteAdmins.pdf>
- **dCache documents (experts)**
 - http://www.dcache.org/manuals/experts_docs.shtml
- **dCache Papers/Presentation**
 - <http://www.dcache.org/manuals/index.shtml>
- **dCache workshop 2005**
 - <http://www.dcache.org/downloads/dCache-Workshop.html>
- **The Perfectly Normal File System**
 - <http://www-pnfs.desy.de/>
- **Old DESY dCache web site**
 - <http://www-dcache.desy.de/>



Useful Document links from dCache sites

- **USATLAS dCache system at BNL**
 - <http://www.atlasgrid.bnl.gov/dcache/manuals/>
- **USATLAS dCache tier 1 & 2 dCache systems**
 - http://www.atlasgrid.bnl.gov/dcache_admin/
- **USCMS Tier1 dCache Status (FNAL) <http://cmsdca.fnal.gov/>**
- **CDF Run II dCache System Status (FNAL) <http://cdfdca.fnal.gov/>**
- **GRIDPP dCache documents <http://wiki.gridpp.ac.uk/wiki/DCache>**
- **D-Cache SRM HOWTO**
 - <http://www.ph.ed.ac.uk/~aearl/dcache2/RAL/index.html>
 - <http://www.ph.ed.ac.uk/~aearl/dcache2/dcache/>
 - <http://storage.esc.rl.ac.uk/documentation/html/D-Cache-Howto/>
- **UK dCache experiences FAQ**
<http://www.gridpp.ac.uk/deployment/admin/dcache/faq.html>
- **Dcache Administrator's Guide**
http://www-zeus.desy.de/~fricke/work/dcache/dcache_admin_guide.html